

SFMTA Real-time data requirements for stationless emerging mobility services

8/29/18

This document describes the SFMTA's real-time data reporting requirements for stationless permit programs (e.g., Stationless Bicycle Share, Powered Scooter). The SFMTA may request additional information in other forms (e.g., summary monthly report, survey of users) as part of the terms and conditions of the permit program.

Collecting this data (in addition to other periodic data reporting requirements) will enable the SFMTA to collect data that will support:

- Managing permittees and operating permit programs
- Enforcing permittee's adherence to permit terms and conditions
- Evaluating permit programs
- Collecting data to support planning efforts consistent with the agency's strategic goals for transportation

Success in each area is highly dependent on staff having access to properly structured data in a timely manner. Towards that end, providers will be required to share data with the SFMTA via a set of Application Programming Interfaces (APIs). To minimize development effort and overhead for permittees, the SFMTA will adhere as closely as possible to existing and emerging standards. This includes using the General Bike Feed Specification (GBFS) as well as the emerging Mobility Data Specification (MDS) drafted by the City of Los Angeles.

For current operations, all permittees must implement the GBFS. Additionally, the SFMTA will require permittees include a request parameter allowing the agency to obtain data for specific points in time.

While the SFMTA requires this detailed data to manage the public right-of-way and support the agency's strategic goals, the SFMTA recognizes that some of this data may be considered sensitive and will aggregate data temporally and spatially when sharing this information outside of the agency.

The remainder of this document describes the following API endpoints in more detail, and what information they are intended to capture. Each of these are based on the MDS. The APIs are divided into "provider" and "agency" based on who is responsible for implementing the server



side of a particular API endpoint. For the purposes of this document, a “provider” is a permitted mobility services provider, and “agency” is the SFMTA.

Interchange Protocol

All data shall be transmitted as message bodies for HTTPS requests or replies. The use of encryption is mandatory. Access to every service shall be controlled using HTTP basic authentication. The mechanism for exchanging authentication credentials is not described in this document.

Data Format

- All message bodies shall be valid [JSON](#).
- All message bodies shall be encoded using [UTF-8](#).
- Line breaks are optional, but if present shall be identified with a single newline character (`\n`).

Field Definitions

- Timestamp fields shall be a JSON number containing the number of milliseconds since January 1, 1970 00:00:00 UTC. Leap seconds shall be accounted for using the [UTC-SLS Proposal](#)
- Id fields in the document should be represented as strings that identify that particular object. They:
 - must be unique within like fields for the same provider (device_id must be unique among devices)
 - do not have to be globally unique
 - must not contain spaces
 - should be persistent for a given object (device, area, etc)
- Enumerable values consist of a list of JSON strings. Values in fields of this type should match an item in the list exactly. The list should be expected to change over time. Values will not be removed, but new valid values may be added as business requirements change and consumers should be designed to handle these changes.
- Point fields shall be [GeoJSON Points](#). Coordinate precision is six decimal places.
- LineString fields shall be [GeoJSON LineStrings](#). Coordinate precision is six decimal places.
- MultiPolygon fields shall be [GeoJSON MultiPolygons](#). Coordinate precision is six decimal places.
- Coordinate pair fields shall be a latitude, a literal comma (”,”) and a longitude. Each coordinate shall be specified in decimal degrees with a precision of six decimal places. The reference system shall be the same as is used by [GeoJSON](#). Coordinate pairs are always used



in this document to define bounding boxes by specifying the northwest and southeast corners of a rectangle in the GeoJSON reference system.

- JSON empty object is an open brace followed immediately by the close brace {}.

Mobility Device Types

Name	Description
bicycle	A bicycle powered solely by its rider
electric_bicycle	A bicycle with electrical power assist
electric_scooter	A powered scooter

Provider API Endpoints

- Service Area endpoint – A record for every polygon that defines a service area, and the start/end dates that each area went into/out of effect.
- Trips endpoint – A record for each trip taken, including start/end time/location.
- Historical Status Change endpoint – A record indicating the time/location of each device when its status changed (e.g., becomes reserved, available for use, unavailable for use, or removed from service) Data from these endpoints is intended to be near real-time; permittees should make new data available to the agency as soon as they are available.

Agency API Endpoints

- Service Area endpoint – A record for every polygon that defines a service area, and the start/end dates that each area went into/out of effect.
- Status Change endpoint – A record indicating the time/location of each device when its status changes (e.g., becomes reserved, available for use, unavailable for use, or removed from service)

API Detail

- Types mentioned in this document are the standard JSON types unless otherwise described in *Field Definitions* above.
- The *provider_id* is issued by the agency, and is guaranteed to be unique across providers.

Data Retention Requirement

All the endpoints which provide historical data must do so for at least the previous two years. Each endpoint specifies whether it is subject to his retention requirement. What shall be returned for specific endpoints in the case where the data does not exist back that far is specified for each endpoint where it is applicable.



Error Body

In cases where an error is returned, the response body shall have the following fields

Field Name	Type	Required	Defines
errors	Array	Yes	A list of strings. Must contain at least one error message
-	String	Yes	Informative error message

Example:

```
{
  "errors": [
    "box_se must be specified if box_nw is specified!",
    "end_time happens before start_time!"
  ]
}
```

Provider Endpoints

Service Areas

Service areas are the geographic regions within which a mobility as a service provider operates. This endpoint is subject to the [Data Retention Requirement](#).

Endpoint: `/service_areas`

Method: GET

Request Parameters

Name	Type	Required
start_time	Timestamp	No
end_time	Timestamp	No

These two parameters define a time period for which service areas are to be described. The current service areas shall be returned if neither parameter is specified. Time periods in the future are allowed. The current service areas shall be returned if there are no changes foreseen for a period in the future. If a time period in the past starts or occurs entirely before the earliest version of all service areas, the first set of areas by chronological order shall be returned. Both or neither parameters should be specified. A response with status code 400 and an error body shall be returned for requests with just one of these parameters specified and for requests that specify an end_time which is smaller than the start_time.

Response body:

Field Name	Type	Required	Defines
provider_id	Id	Yes	
areas	Array	Yes	A list of area objects as defined below
- area_id	Id	Yes	Unique identifier of a service area.
- start_date	Timestamp	Yes	Date at which this service area became effective.
- end_date	Timestamp	Yes	Date at which this service area was replaced. Omit if it's the current effective area of this type
- area	MultiPolygon	Yes	GeoJSON MultiPolygon for this area
- prior_area_id	Id	No	If exists, the id of the prior service area for this type. See types below

- replacement_area_id	Id	No	If exists, the id of the area that replaced this type. See types below
- type	Enumerable	Yes	One of seven types that describes the intent of the service area geography. See types below

Area types

Name	Description	Notes
unrestricted	Areas where devices may be picked up/dropped off	A provider's unrestricted area shall be contained completely inside the agency's unrestricted area for the provider in question, but it need not cover the entire agency unrestricted area. See the agency version of the service areas endpoint below
agency_restricted	Areas where the agency does not allow device pick-up/drop-off	
provider_restricted	Areas where the provider does not allow device pick-up/drop-off	
agency_preferred_pick_up	Areas where users are encouraged by the agency to pick up devices	
provider_preferred_pick_up	Areas where users are encouraged by the provider to pick up devices	
agency_preferred_drop_off	Areas where users are encouraged by the agency to drop off devices	
provider_preferred_drop_off	Areas where users are encouraged by the provider to drop off devices	



the provider to
drop off devices

Trips

A trip represents a journey taken by a mobility as a service customer with a geo-tagged start and stop point. The trips API endpoint shall be queriable for historical trip data. The endpoint should allow querying trips at least by *device_id*, geographical bounding box, and time. If a geographical bounding box is specified, any trip that has a point in the bounding box shall be returned. This endpoint is subject to the [Data Retention Requirement](#).

Endpoint: `/trips`

Method: GET

Request Parameters

Name	Type	Required
start_time	Timestamp	Yes
end_time	Timestamp	Yes
device_id	Id	No
box_nw	Coordinate pair	No
box_se	Coordinate pair	No

The `box_*` parameters must be specified together or not at all. A provider shall return a response with status code 400 and an error body for requests with just one of these parameters specified. Trips for the entire service area shall be returned if a bounding box is not specified. Trips for all devices shall be returned if a device Id is not specified. The provider shall return the empty JSON object for requests with an `end_time` that happened before the earliest available trip. A response with status code 400 and an error body shall be returned for requests with a `start_time` in the future or an `end_time` which is smaller than the `start_time`.

Response body:

Field Name	Type	Required	Defines
provider_id	Id	Yes	
trips	Array	Yes	A list of trip objects as defined below
-device_type	String	Yes	
-device_id	Id	Yes	
-trip_id	Id	Yes	
-duration	Number	Yes	Time, in Seconds



-distance	Number	Yes	Distance, in Meters
-start_point	Point	Yes	
-end_point	Point	Yes	
-accuracy	Number	Yes	The approximate level of accuracy for start_point and end_point, in meters.
-route	LineString	Yes	
-sample_rate	Number	Yes	The frequency, in seconds, in which the route is sampled.
-start_time	Timestamp	Yes	
-end_time	Timestamp	Yes	
-membership_type	Enumerable	Yes	Membership type for the user.
-device_occupancy	Number	No	Capture vehicle occupancy (n/a for scooters and bikes)
-standard_cost	Number	Yes	The cost, in cents that it would cost to perform that trip in the standard operation of the System.
-actual_cost	Number	Yes	The actual cost paid by the user of the Mobility as a service provider

Membership types

Name	Description
subscriber	
subscriber_low_income	
single_ride	
single_ride_low_income	

Status Change

Status changes for mobility devices. This endpoint is subject to the [Data Retention Requirement](#).

Endpoint: `/device_status`

Method: GET

Request Parameters

Name	Type	Required
------	------	----------



utc_hour	Timestamp	Yes
device_id	Id	No
box_nw	Coordinate pair	No
box_se	Coordinate pair	No

The `box_*` parameters must be specified together or not at all. A provider shall return a response with status code 400 and an error body for requests with just one of these parameters specified. Status changes for the entire service area shall be returned if a bounding box is not specified. Status changes for all devices shall be returned if a device Id is not specified. The `utc_hour` parameter shall be the timestamp for the first millisecond of an hour in the UTC timezone. Status changes for the following hour shall be returned. The provider shall return the empty JSON object for requests with an `utc_hour` that happened before the earliest available trip. A response with status code 400 and an error body shall be returned for requests with a `utc_hour` in the future. A provider may choose to truncate the timestamp value to the previous hour for requests with a `utc_hour` parameters which does not properly correspond to the beginning of a UTC hour, or may choose return a response with status code 400 and an error body for such requests.

Response body:

Field Name	Type	Required	Defines
provider_id	Id	Yes	
device_status	Array	Yes	A list of device status objects as defined below
-device_id	Id	Yes	
-device_type	Enumerable	Yes	See Mobility Device Types above
-event_type	String	Yes	Four types. Described in Appendix A
-reason	String	Yes	Reason for status change. Described in Appendix A
-time	Timestamp	Yes	When the event occurred
-position	Point	Yes	Event location
-battery_pct	Number	Yes	Percent battery charge of device, expressed as a fraction between 0 and 1. Specify 0 for unpowered devices
-trip_id	Id	Yes	Required for “Reserved” event types, associated trip. Details should be available using the provider Trips API endpoint

Agency Endpoints

Service Areas

Service areas are the geographic regions within which a mobility as a service provider is permitted to operate. This endpoint is subject to the [Data Retention Requirement](#).

Endpoint: `/service_areas`

Method: GET

Request Parameters

Name	Type	Required
start_time	Timestamp	No
end_time	Timestamp	No

These two parameters define a time period for which service areas are to be described. The current service areas shall be returned if neither parameter is specified. Time periods in the future are allowed. The current service areas shall be returned if there are no changes foreseen for a given period in the future. If a time period in the past starts or occurs entirely before the earliest version of all service areas, the first set of areas by chronological order shall be returned. Both or neither parameters should be specified. A response with status code 400 and an error body shall be returned for requests with just one of these parameters specified and for requests that specify an end_time which is smaller than the start_time.

Response body:

Field Name	Type	Required	Defines
provider_id	Id	Yes	
areas	Array	Yes	A list of area objects as defined below
- area_id	Id	Yes	Unique identifier of a service area.
- start_date	Timestamp	Yes	Date at which this service area became effective.
- end_date	Timestamp	Yes	Date at which this service area was replaced. Omit if it's the current effective area of this type
- area	MultiPolygon	Yes	GeoJSON MultiPolygon for this area
- prior_area_id	Id	No	If exists, the id of the prior service area for this type. See types below



- replacement_area_id	Id	No	If exists, the id of the area that replaced this type. See types below
- type	String	Yes	One of four types that describes the intent of the service area geography. See types below

Area types

Name	Description	Notes
unrestricted	Areas where devices may be picked up/dropped off.	A provider may choose to operate in a subset of this area. However, all of the provider's operating area must be completely contained in this operating area
agency_restricted	Areas where the agency does not allow device pick-up/drop-off	
agency_preferred_pick_up	Areas where users are encouraged by the agency to pick up devices	
agency_preferred_drop_off	Areas where users are encouraged by the agency to drop off devices	

Status Change

The Status Change endpoint allows providers to communicate the location for all devices on the street when their status changes (e.g., a user completes a reservation and the device is now available). When a device's status changes, the provider will push one of four event types with additional descriptive elements.

Endpoint: `/device_status`

Method: POST

Response body:

Field Name	Type	Required	Defines
provider_id	Id	Yes	

device_id	Id	Yes	
device_type	Enumerable	Yes	See Mobility Device Types above
event_type	String	Yes	Four types. Described in Appendix A
reason	String	Yes	Reason for status change. Allowable values determined by event_type. Described in Appendix A
time	Timestamp	Yes	When the event occurred
position	Point	Yes	Event location
battery_pct	Number	Yes	Percent battery charge of device, expressed as a fraction between 0 and 1. Specify 0 for unpowered devices
trip_id	Id	Yes	Required for “Reserved” event types, associated trip. Details should be available using the provider Trips API endpoint

Appendix A: Status Change Event Types and Allowable Values for Reason

event_type	event_type description	reason	reason description
available	A device becomes available for customer use	service_start	Device introduced into service at the beginning of the day (if program does not operate 24/7)
		user_drop_off	User ends reservation
		rebalance_drop_off	Device moved for rebalancing
		maintenance_drop_off	Device introduced into service after being removed for maintenance
reserved	A customer reserves a device (even if trip has not started yet)	user_pick_up	Customer reserves device
unavailable	A device is on the street but becomes unavailable for customer use	maintenance_user	A device is no longer available due to equipment issues – initiated by a user
		maintenance_provider	A device is no longer available due to equipment issues – initiated by provider

		low_battery	A device is no longer available due to insufficient battery
removed	A device is removed from the street and unavailable for customer use	service_end	Device removed from street because service has ended for the day (if program does not operate 24/7)
		rebalance_pick_up	Device removed from street and will be placed at another location to rebalance service
		maintenance_pick_up	Device removed from street so it can be worked on

Allowable event_type transitions

- removed->available
- available->reserved
- available->unavailable
- available->removed
- reserved->available
- reserved->unavailable
- unavailable->available
- unavailable->removed